



Interrogation de données en MongoDB

1. Filtrage de données:

- Le filtrage de données dans MongoDB est une opération fondamentale qui vous permet de récupérer des documents de votre collection en fonction de certains critères.
- La méthode **find()** est utilisée pour rechercher des documents dans une collection.
- **Exemple :**

```
db.MyCollection.find({ "nom": "Mohamed" })
```

```
db.MyCollection.find({"salaire":{$gt:5000}});
```

1. Filtrage de données :

- En MongoDB, les expressions régulières sont utilisées pour effectuer des requêtes de recherche basées sur des motifs dans les chaînes de caractères. Vous pouvez utiliser l'opérateur **\$regex** pour appliquer une expression régulière dans vos requêtes.

- **Exemple :**

```
db.MyCollection.find({"nom":{"$regex:/ali/}});
```

```
db.MyCollection.find({"nom":{"$regex:/^a.*i$/}});
```

2. Projection:

- La projection vous permet de spécifier quels champs vous souhaitez inclure ou exclure des résultats
- **Exemple :**

```
db.MyCollection.find({}, {"nom":1,"prenom":1});
```

```
db.MyCollection.find({"salaire":{"$gt:5000"}}, {"nom":1,"prenom":1,"_id":0});
```

- **Remarque:**

On peut créer les deux documents JSON de la sélection et la projection en dehors de la requête

```
selection = {"salaire":{"$gt:5000"}}  
projection = {"nom":1,"prenom":1,"_id":0}  
db.MyCollection.find(selection,projection);
```

3. Tri des données:

- Le tri des documents renvoyés par une fonction `find()` est réalisable avec la fonction **`sort()`**.
- On doit indiquer les champs de tri et leur attribuer une valeur de 1 pour un tri ascendant et une valeur de -1 pour un tri descendant,
- **Exemple :**

```
db.MyCollection.find({}).sort({"nom":1});
```

```
db.MyCollection.find({"salaire":{"$gt":5000}}).sort({"nom":1,"prenom":-1});
```

4. Limiter les données:

- **limit(n)** s'ajoute à la suite d'une fonction find() et retourne les n premiers documents résultats
- Cette fonction permet de limiter le nombre de documents renvoyés par la fonction find().
- **Exemple :**

```
db.MyCollection.find({}).limit(5);
```

```
db.MyCollection.find({"salaire":{$gt:5000}}).sort({"nom":1,"prenom":-1}).limit(3);
```


5. Compter les données:

- La fonction `count()` s'ajoute à la suite d'une fonction `find()` et retourne le nombre de documents renvoyés
- **Exemple :**

```
db.MyCollection.find({}).count();
```

```
db.MyCollection.find({"salaire":{$gt:5000}}).count();
```

6. DISTINCT:

- La fonction **DISTINCT** en MongoDB est utilisée pour récupérer les valeurs distinctes d'un champ spécifique dans une collection. Cela peut être utile lorsque vous souhaitez connaître toutes les valeurs uniques présentes dans un champ particulier.
- **Exemple :**

```
db.MyCollection.distinct("ville")
```




TP8